05

Para saber más: hash de contraseña

Al implementar una funcionalidad de autenticación en una aplicación, independientemente del lenguaje de programación utilizado, deberá tratar con los datos de inicio de sesión y contraseña de los usuarios, y deberán almacenarse en algún lugar, como, por ejemplo, una base de datos.

Las contraseñas son información confidencial y no deben almacenarse en texto sin formato, ya que si una persona malintencionada logra acceder a la base de datos, podrá acceder a las contraseñas de todos los usuarios. Para evitar este problema, siempre debe usar algún algoritmo hash en las contraseñas antes de almacenarlas en la base de datos.

Hashing no es más que una función matemática que convierte un texto en otro texto totalmente diferente y difícil de deducir. Por ejemplo, el texto “Mi nombre es Rodrigo” se puede convertir en el texto 8132f7cb860e9ce4c1d9062d2a5d1848, utilizando el algoritmo hash MD5.

Un detalle importante es que los algoritmos de hash deben ser unidireccionales, es decir, no debe ser posible obtener el texto original a partir de un hash. Así, para saber si un usuario ingresó la contraseña correcta al intentar autenticarse en una aplicación, debemos tomar la contraseña que ingresó y generar su hash, para luego compararla con el hash que está almacenado en la base de datos.

Hay varios algoritmos hashing que se pueden usar para transformar las contraseñas de los usuarios, algunos de los cuales son más antiguos y ya no se consideran seguros en la actualidad, como MD5 y SHA1. Los principales algoritmos actualmente recomendados son:

- Bcrypt

- Scrypt
- Argon2
- PBKDF2

A lo largo del curso utilizaremos el algoritmo BCrypt, que es bastante popular hoy en día. Esta opción también tiene en cuenta que Spring Security ya nos proporciona una clase que lo implementa.